**Assembler Implementation Documentation**

Here is the implementation of four pseudocode segments using Hack assembler language. Each pseudocode segment corresponds to a specific problem statement and is implemented using Hack assembly instructions. Below is a brief documentation for each problem section:

**Problem 1: For Loop**

* **Objective**: Implement a for loop decrementing a variable **j** until it reaches 0.
* **Assembler Solution**: Initializes **j** to 5 and iteratively decrements it until it reaches 0.
* **Implementation Details**: Utilizes the JGE instruction for conditional jumping and the M register for storing variables.

**Problem 2: If-Then-Else**

* **Objective**: Implement an if-then-else statement based on the value of variable **i**.
* **Assembler Solution**: Checks if **i** is less than 5, assigns 3 to **j** if true, and otherwise assigns 2.
* **Implementation Details**: Uses JGE instruction for conditional branching and memory registers for variable storage.

**Problem 3: While Loop**

* **Objective**: Implement a while loop incrementing a variable **j** until **i** becomes non-zero, then set **i** to the value of **j** when **j** equals 5.
* **Assembler Solution**: Increments **j** within the loop and checks if **j** equals 5 to exit the loop.
* **Implementation Details**: Employs JE instruction for conditional branching, loop control with memory registers, and memory manipulation for updating variables.

**Problem 4: Load and Traverse an Array**

* **Objective**: Load values into an array and traverse it, replacing any occurrence of 0 with 5.
* **Assembler Solution**: Initializes an array with predefined values and traverses it, replacing 0s with 5s.
* **Implementation Details**: Utilizes memory addressing for array manipulation, conditional branching for traversal, and memory updates for replacing values.